

REMARKS

Applicant gratefully acknowledges the courtesy extended to the undersigned representative during the telephone interview on February 28, 2008, in which the examiner confirmed that the requirement set forth in point 3 on page 2 of the Office Action, although characterized as a requirement for election of species, is in fact a requirement for restriction. In response to the requirement for restriction, applicant elects the subject matter of claims 41-52 and 62.

Claims 45-52 stand rejected under 35 USC 101. Applicant proposes that claim 45 should be amended as indicated above so that it is directed to a storage medium rather than a software protection arrangement. Amendment of claim 45 in the manner indicated above is equivalent to rewriting claim 62 in independent form. Consequential amendments to the dependent claims 46-52 are presented. Applicant believes that these amendments clearly remove the rejection of claims 45-52 under 35 USC 101. Applicant requests that the amendments be entered.

Claims 41-52 and 62 stand rejected under 35 USC 103 over Shimizu et al in view of Hughes et al, Mittal et al and Yeung et al.

There are at least two important distinctions between the subject matter of claim 41 and the prior art:

1. The server of claim 41 verifies that use of the protected software by the wireless device is authorized before generating a derived identifier to send to the device.
2. A derived identifier which is formed from a function operating on an identifier and a decryption key is transmitted from the server to the wireless device. The wireless device performs a function on the derived identifier, using the identifier of the device, to recover the decryption key. The decryption key is then used to decrypt software on the device.

As discussed in the reply to the previous Office Action, two conditions must be met in order for the protected software to run on the wireless device. First, in accordance with point 1 above, the server must verify that use of the protected software is authorized (for example by operating on the identifier with a function and comparing the result returned by the operation with a value stored in the server). Only if the result of the authorization is positive does

the server then form the derived identifier (point 2), which is sent to the wireless device to recover the decryption key. If the wireless device is able to recover the proper decryption key, the encrypted software is decrypted at the wireless device and can then run on the wireless device.

The subject matter of claim 41 only allows a decryption key to be retrieved by a wireless device if the server verifies that the wireless device is authorized. Therefore, the software can only be decrypted on a wireless device that has been authorized by the server.

Shimizu et al

Shimizu et al requires a user to register in advance with a file server (column 4, lines 36-39) to receive a registration number. In use, the device transmits the registration number together with a request for an ID number to a file server (column 4, lines 41-43). The server verifies that the user is authorized to receive the ID number by examining the registration number. If the user is authorized then the server transmits the ID number to the device (column 4, lines 43-47). The device then transmits the received ID number to a key server and requests transmission of a decryption key (column 4, lines 48-51). The key server uses the ID number to retrieve a decryption key and then transmits the decryption key to the device (column 4, lines 52-56). The device then requests the encrypted software from the file server and uses the decryption key to decrypt the software and execute the software (column 4, lines 57-65).

The server in Shimizu et al does not perform a function on the decryption key to form a derived identifier which is then transmitted to the device. Moreover, Shimizu et al does not disclose that a derived identifier is formed from a function being performed on the decryption key and an identifier received from the device.

The system disclosed in Shimizu et al is not secure because the system does not employ encryption to protect the decryption key, but instead relies on a manager or another authorized person to distribute software legitimately under a licensing agreement (column 2, lines 47-53 and column 2, line 61-62).

Hughes et al

Hughes et al discloses a system in which a hardware ID is generated (para. 36) and then sent to an activation server (para. 40). The activation server generates a license file by performing a function on the hardware ID and a product ID. The license file is then sent to the device and stored on the device (para. 41). The device then performs the same function as that performed by the server on the product ID and the hardware ID to produce a test file (para. 43). The device compares the test file with the license file and, if the files match, the device allows software to run.

The license file which is generated by the server in Hughes et al is not a derived identifier within the meaning of claim 41 because it is not formed from a function being performed on a decryption key and an identifier received from the device. The device in Hughes carries out a simple comparison between the license file and a test file which has been generated using the same function. The result of this simple comparison is the criterion used to determine whether software is authorized to run. Hughes et al does not disclose a system in which a decryption key is retrieved from a derived identifier to decrypt software.

Mittal et al

Mittal et al discloses a method for confirming the identity of a computer system to a server (column 2, lines 21-23). To confirm the identity of a computer system a processor in the computer system compares an expected hash value with a generated hash value (column 3, lines 32-37 and FIG. 2). The expected hash value may be derived from a key and a first identifier for a computer system and the generated hash value may be derived from that key and a second identifier for a computer system (column 3, lines 34-37). If the generated hash value matches the expected hash value then the method generates a "true response" which indicates that the computer system currently executing an application is identical to the computer system that is authorized to execute the application (column 4, lines 48-51).

The method disclosed in Mittal et al involves the decryption of data by a decryption program (column 5, lines 24-26). However, the "keys" described in Mittal et al are not used as decryption keys in the decryption program.

The "keys" referred to in Mittal et al are strings that correspond to a web address or URL (column 1, lines 36-65). The "keys" disclosed in Mittal et al are not decryption keys within the meaning of claim 41 because the keys are not combined with a decryption algorithm to decrypt encrypted software as is the case in the subject matter of claim 41.

Mittal et al does not disclose a decryption key of any kind. Consequently, Mittal et al does not teach or even suggest forming a derived identifier by performing a function on a hardware identifier and a decryption key, with the derived identifier being transmitted from a server to a wireless device.

Yeung et al

Yeung et al discloses a system in which a server generates keys by performing a function on a hardware ID (CPU_ID) and a recipient identifier (REC_ID) (column 6, lines 9-19). The keys are generated from "client-based information", as indicated in FIG. 2. The keys are then used to scramble or encrypt data (column 6, lines 32-37). The encrypted data is then sent to a client device, where it is stored in a memory unit (FIG. 4 and column 7, lines 36-37). The client device then performs a function on the hardware ID (CPU_ID) and the other auxiliary information, such as the recipient ID (REC_ID), to replicate the keys that were produced originally on the server (column 7, lines 37-41). Once the client device has generated the keys the client device can use the keys to descramble or decrypt the data,

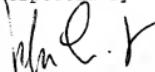
The system disclosed in Yeung et al relies on a client device to generate decryption keys from client-based information. Yeung et al does not teach or even suggest forming a derived identifier by performing a function on a decryption key and an identifier received from a device. Yeung et al does not teach or even suggest transmitting a derived identifier to a wireless device for the wireless device to retrieve the decryption key by performing a function on the derived identifier and a device identifier.

The method of claim 41 provides a higher level of security than any of the systems disclosed or suggested by the cited references. The prior art systems do not disclose or suggest linking a unique decryption key with a device identifier.

In view of the foregoing, applicant submits that even though the examiner has found it necessary to cite four prior art documents in support of the rejection of claim 41, the cited documents do not in fact justify a rejection of claim 41 because they do not disclose or suggest the two important distinctions mentioned above. Therefore, claim 41 is patentable and it follows that the dependent claims 42-44 also are patentable.

The arguments presented in support of claim 41 are applicable to claim 45 also because claim 45 contains equivalent limitations to claim 41. Therefore claim 45 is patentable and it follows that the dependent claims 46-52 also are patentable.

Respectfully submitted,



John Smith-Hill
Reg. No. 27,730

SMITH-HILL & BEDELL, P.C.
16100 N.W. Cornell Road, Suite 220
Beaverton, Oregon 97006

Tel. (503) 574-3100
Fax (503) 574-3197
Docket: FORR 2793